SIPEW 2008
SPEC International Performance Evaluation Workshop
June 27-28, 2008
Darmstadt, Germany

Model-Driven Performability Analysis of Composite Services

Paolo Bocciarelli, Andrea D'Ambrogio

Dept. of Computer Science, Systems and Production
University of Roma “Tor Vergata”
Roma (Italy)

Outline

- Performability Analysis in a SOA context
- QoS-Enabled WSDL (Q-WSDL)
- Model-Driven Performability Prediction
  - Performance prediction
  - Reliability prediction
  - Combined Approach: Performability prediction
- Example application
- Conclusions

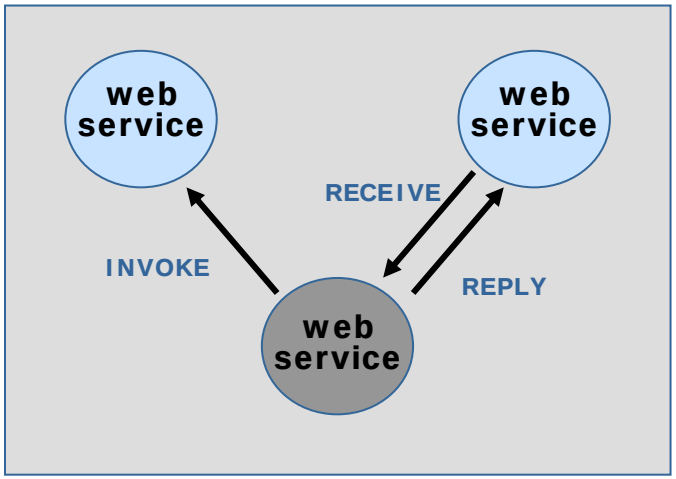
Services Oriented Architectures

- Distributed applications are rapidly converging towards the adoption of a computing paradigm based on *service-oriented architectures* (SOA).
- The service-oriented architecture provides the necessary support for the consolidation of multiple services into a single *composite service* corresponding to the overall process.
- Service providers are interested to describe the *QoS characteristics* of offered services, specifically with regards to the performance and the reliability.
- An approach is needed to *integrate* the performance and the reliability prediction into services composition processes

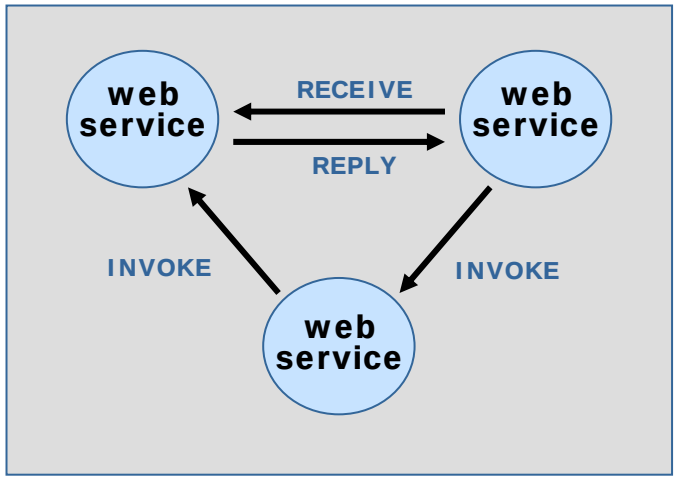
Composite Services

- A composite service can be seen as a set of services that cooperate to execute a process that defines the interaction workflow.

- Different paradigm of collaboration:



ORCHESTRATION



CHOREOGRAPHY

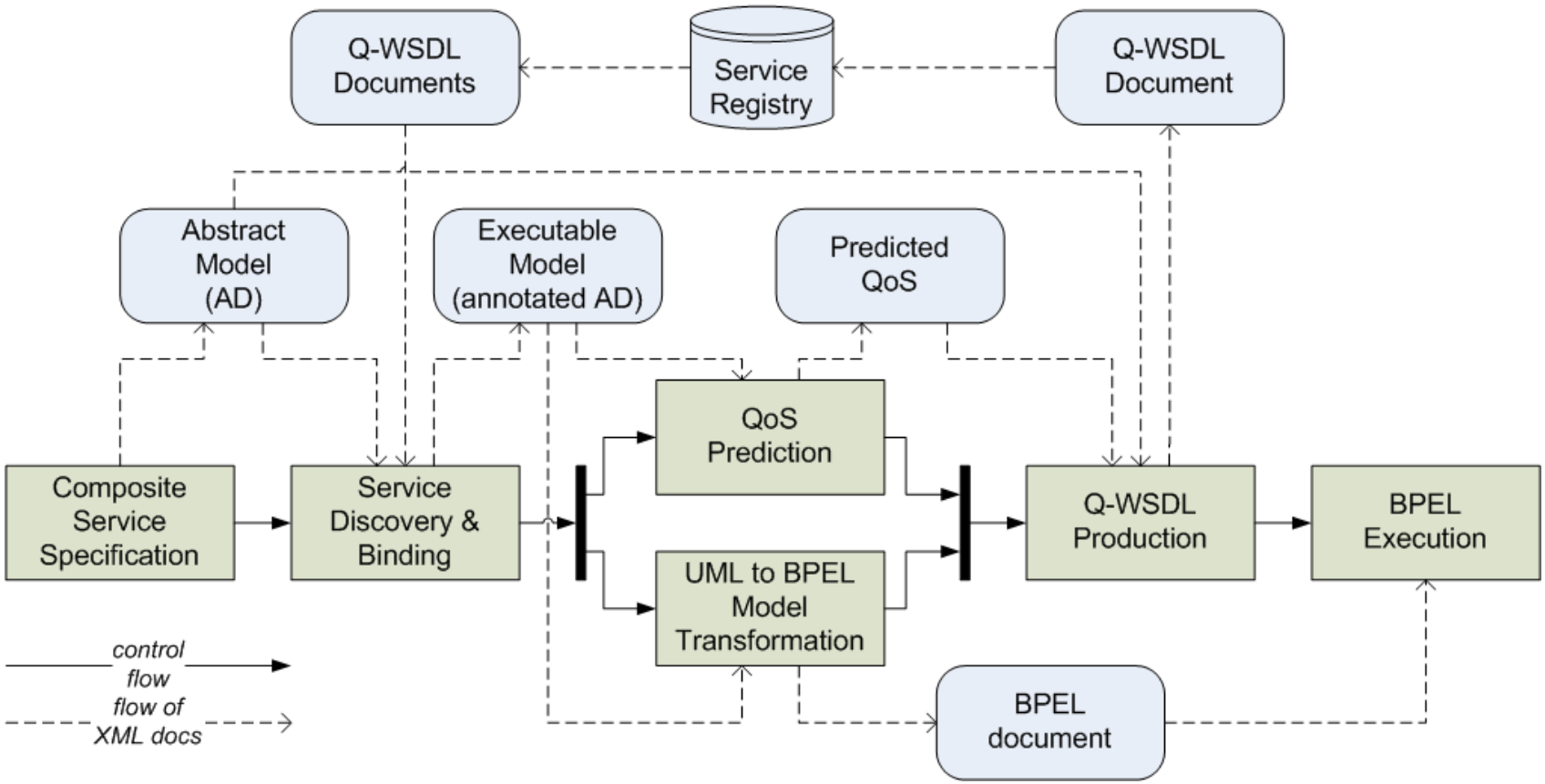
- Several languages was proposed to describe interactions between web servces. We specifically focus on BPEL

QoS-Enabled WSDL

- A WSDL description is an XML document that contains all the information about service capabilities and invocation mechanisms
- A WSDL does not contain any description of QoS characteristics of Web Service
- A WSDL extension is needed
- The proposed (lightweight) extension is based on a metamodel trasformation
- The WSDL and Q-WSDL metamodels are defined by use of the Meta Object Facility (MOF)

Performability Prediction

The proposed method is integrated into a model-driven service composition process



QoS Prediction

- In this work the QoS prediction is specifically carried out in terms of performability (joint analysis of performace and reliability)
- Performance predictions are achieved by use of LQN (Layered Queueing Network) models.
- The Reliability prediction is obtained from the MTTF values specified in Q-WSDL descriptions of component services.
- An algorithm is proposed in order to determine a "Reward Rate" for each candidate configuration, that takes into account both performance and reliability

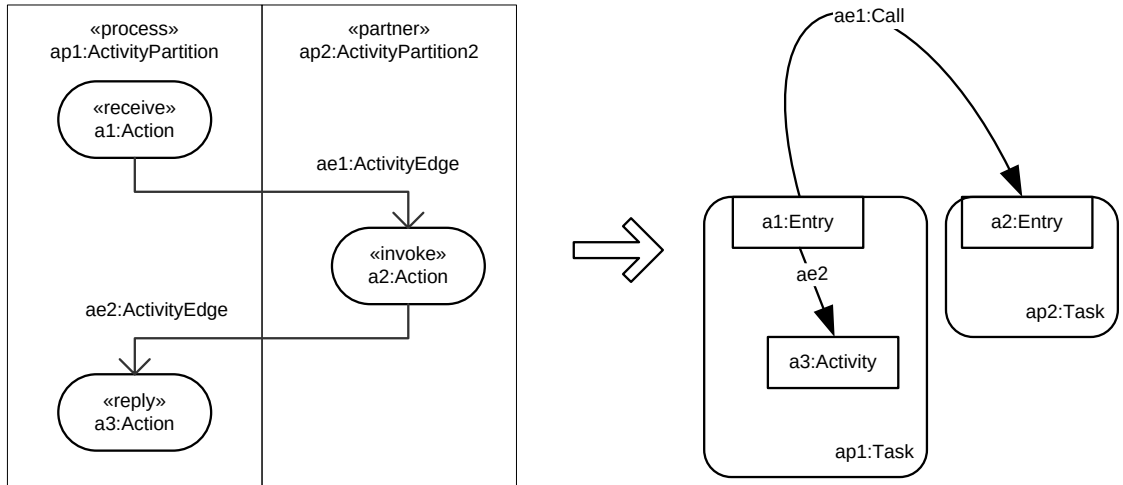
Performance Prediction: BPEL Executable Model

- The annotated AD representing the BPEL executable process, is translated into a LQN model
- The stereotypes introduced to extend UML Activity Diagram:

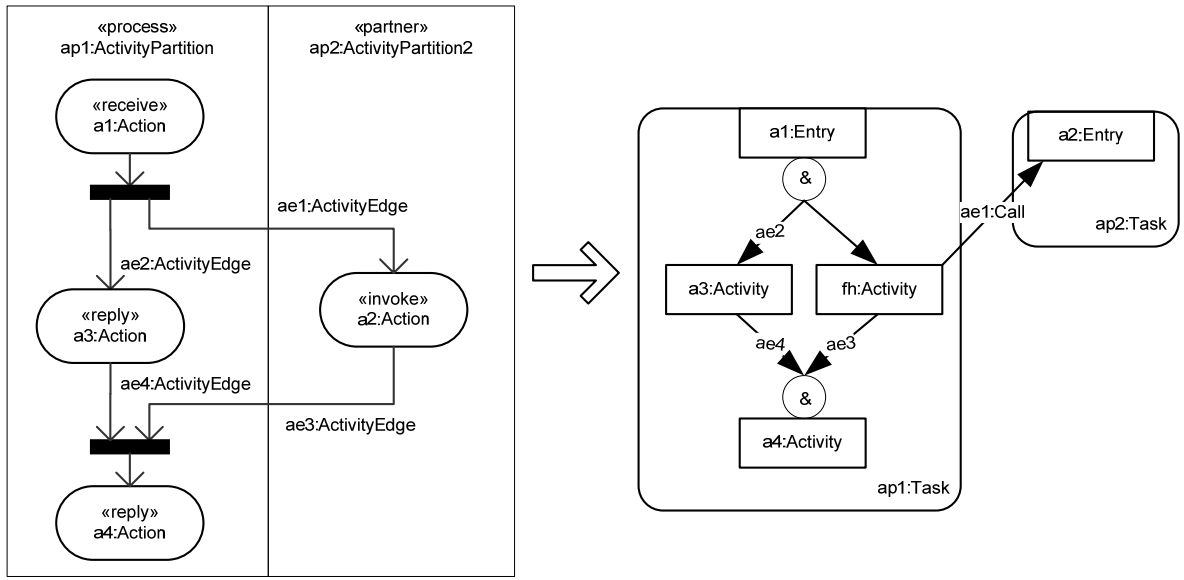
Stereotype	Base Metaclass	Description
«process»	ActivityPartition	BPEL process coordinator
«partner»	ActivityPartition	BPEL partner (component web service)
«receive»	Action	BPEL receive activity
«invoke»	Action	BPEL invoke activity
«reply»	Action	BPEL reply activity

Performance Prediction: Mapping of BPEL into LQNM

Basic Interaction

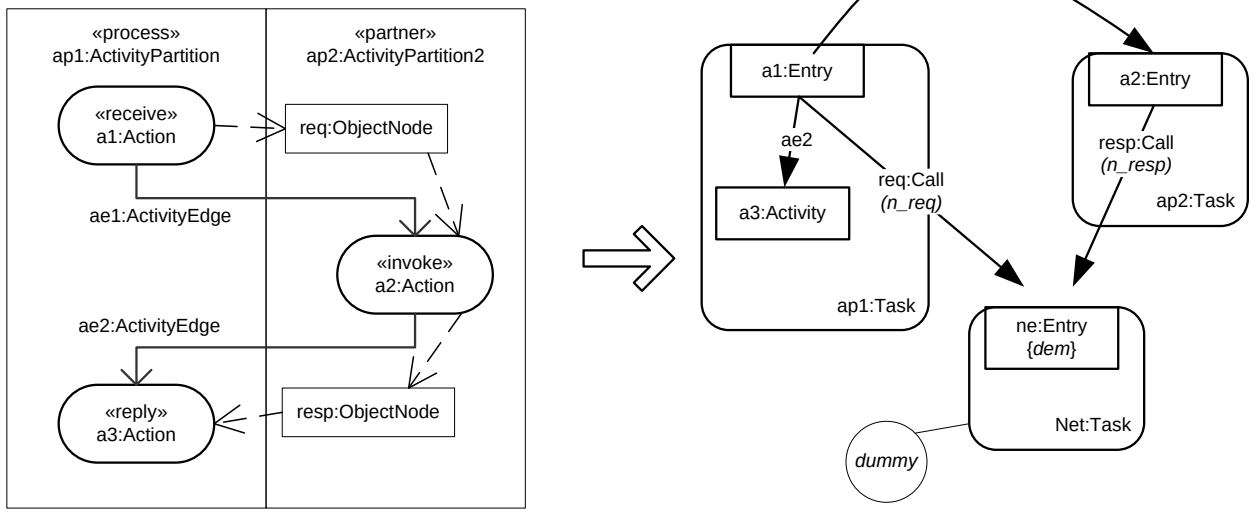


Flow



Performance Prediction: Mapping of BPEL into LQNM

Message Exchange



Conditional and Iterative Primitives

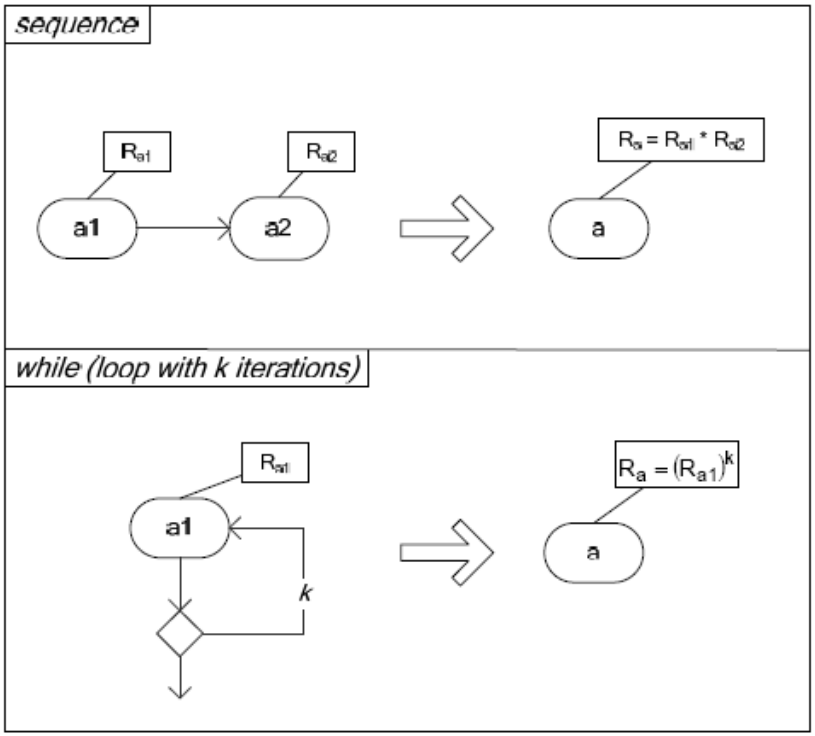
- The transformation of BPEL ***switch*** and ***while*** constructs are dealt by associating PAprob/PArep tagged values to Action instances stereotyped as «PAstep», according to the SPT profile.
- Tagged values are used to compute the number of calls to LQN entries.

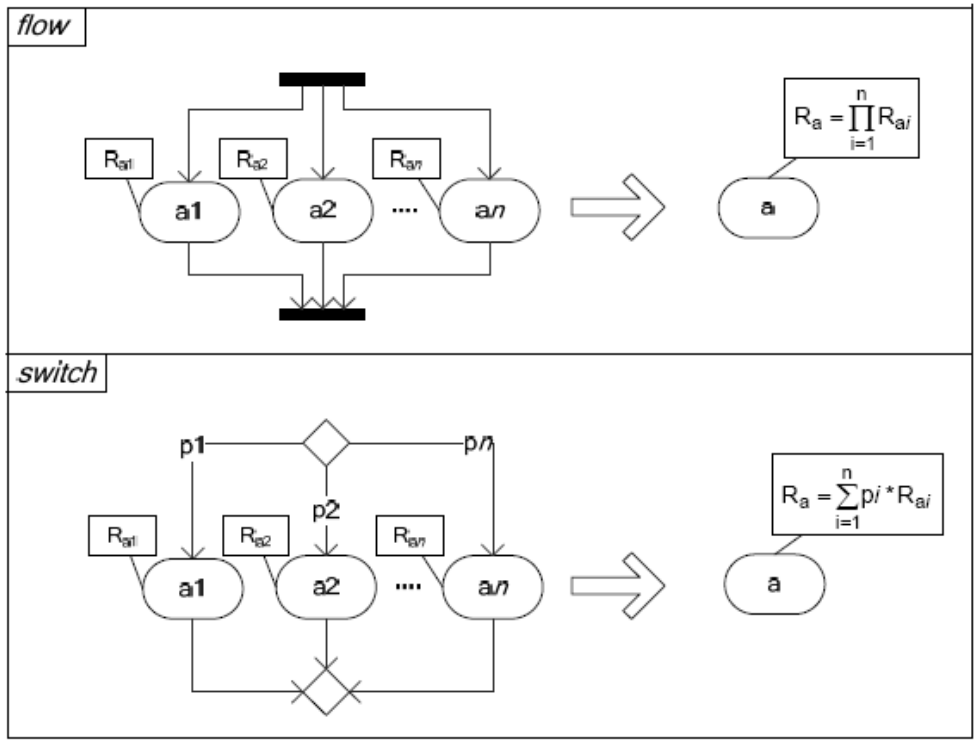
Reliability Prediction

- It is assumed that the MTTF value is specified for each services provided by BPEL partners involved in the composite service.
- By assuming an exponential distribution probability for failures, the reliability associated to each AD node *a (*BPEL basic activity) can be computed as:

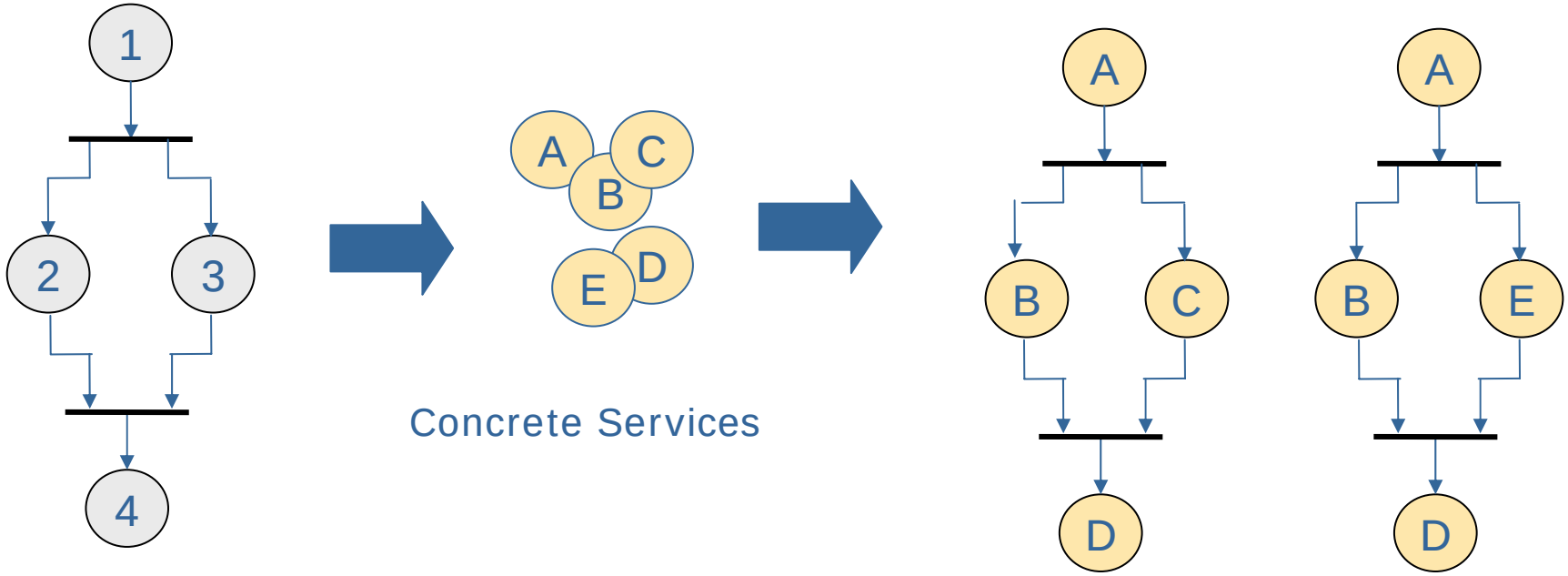
$$R_a(t) = e^{-\frac{1}{MTTF_a}t}$$

Reliability Prediction: Reduction Rules





Performability Prediction: the idea



Abstract Process

Concrete Services

Candidate Configurations

For each (initial) candidate configuration a reward rate is computed: an overall attribute that combines both the throughput and the reliability of the composite service

Performability Prediction: The Algorithm

1. Generate the State Transition Diagram (STD)
2. Select a candidate configuration as the initial configuration
3. Use the reliability predictions to obtain the transition probabilities of the STD
4. Calculate the absorbing probabilities $P(CS_i)$ of being in a given working configuration (i=1..n) starting from the initial configuration;
5. Use the performance predictions to obtain the performance (in terms of the throughput) $T(CS_i)$ associated to each configuration, and assign it as a reward
6. Obtain the performability prediction in terms of the expected reward rate of the composite web service CS given by:

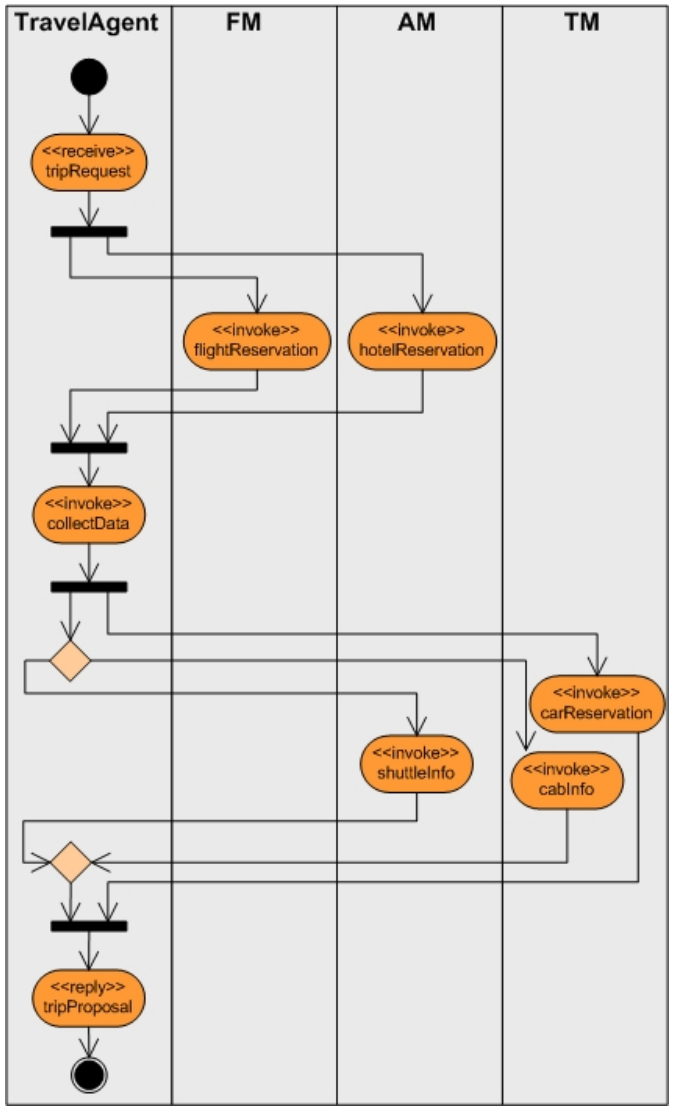
$$RW(CS) = \sum_{i=1}^{n} P(CS_i)\, T(CS_i)$$

Case Study

- Let us consider a composite web service that provides an operation for creating travel plans. The process requires the following services:
  - Flight Manager (FM) service
  - Accommodation Manager (AM) service
  - Transportation Manager (TM) service

- Our intention is to show how the performability analysis can lead to results unexpected if performance and reliability attributes are dealt with separately

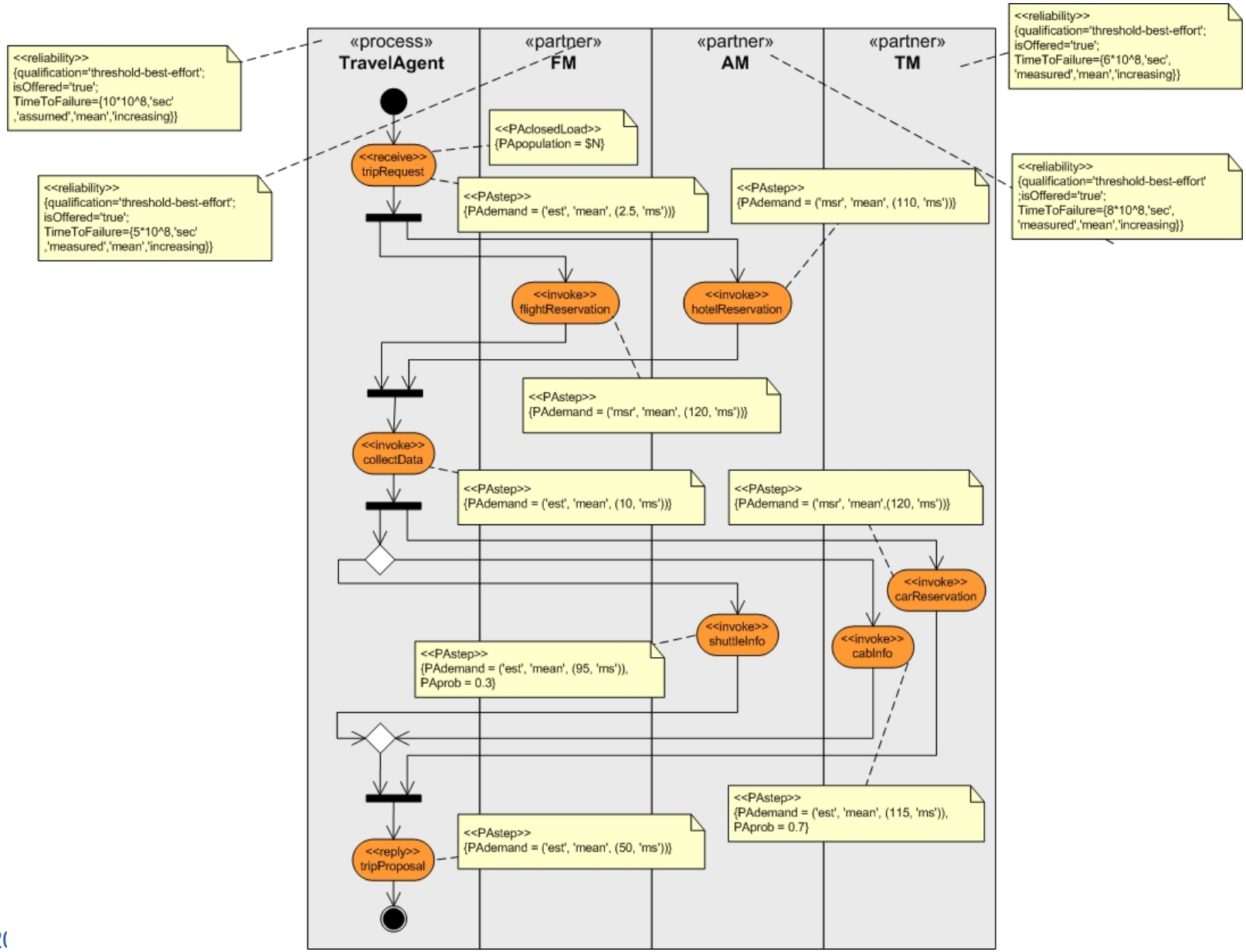
Abstract Model



- The abstract model of the composite web servie is built as a first step of the model-driven performability analysis
- Let us suppose that two different candidate services are available to bind the TM service.

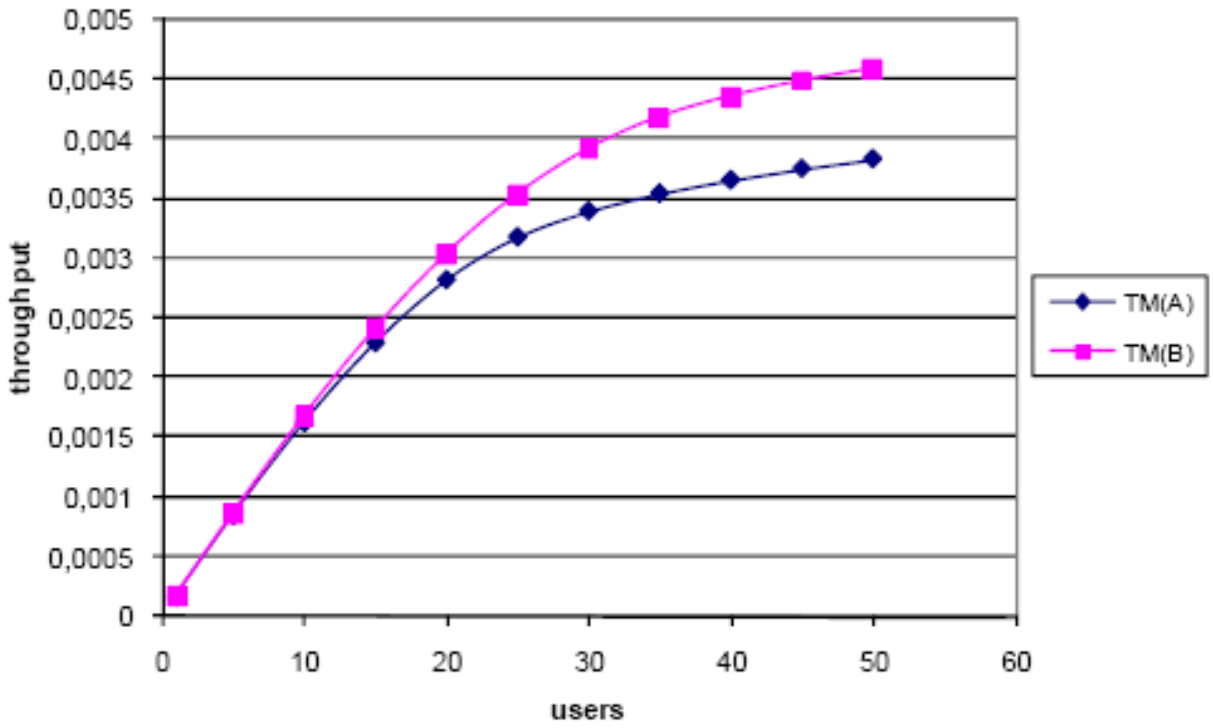
Parameter		TM_A	TM_B
Performance	CarReservation time demand	120 ms	90 ms
	CabInfo time demand	115 ms	84 ms
	Network bit rate	10 Mb	100 Mb
Reliability	MTTF	10*10^8	7.2*10^7
	R(1year)	0.961	0.645

Annotated AD Model



Performance Prediction

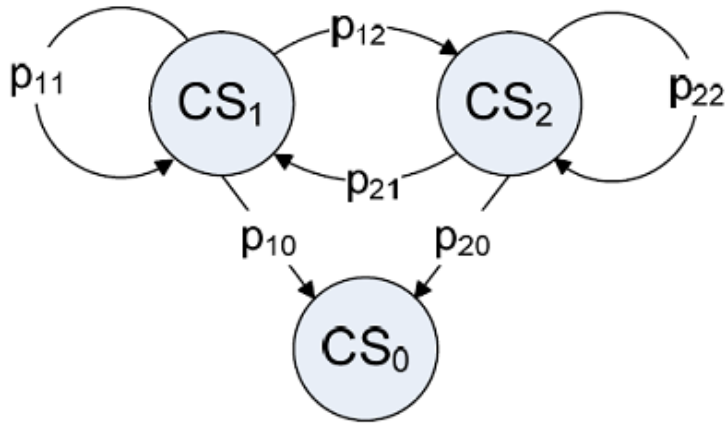
If the prediction activity is limited to performance-related attributes, the choice of the alternative denoted as TM_B is to be preferred as initial configuration for composite service



Perfomability Prediction: Step1: Definition of STD

- The State Transition Diagram represents the possible configurations that the composite service may undergo before experimenting a failure
- The states represent the working configurations, while the transitions represent the probabilities to remain in a configuration or move to a different one.

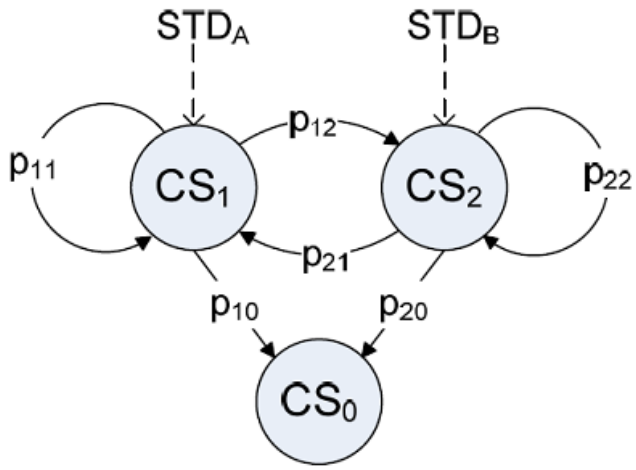
State	Description
CS_0	Composite Service in the failed state
CS_1	Composite Service including TM_A
CS_2	Composite Service including TM_B



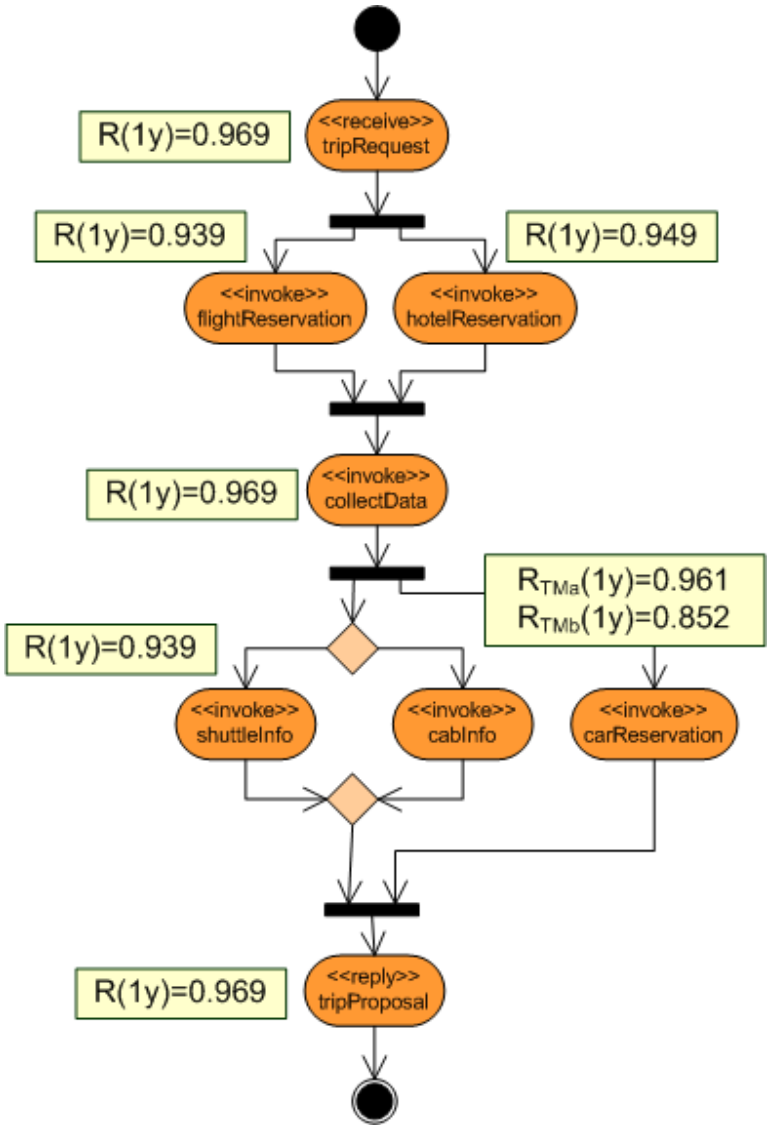
Perfomability Prediction: Step2-3: Transition Probabilities

- At the second step of the algorithm, a candidate initial configuration is selected (two alternative STD_A and STD_B)
- At the third step, the transition probabilities in the STD are obtained by applying the reliability prediction method

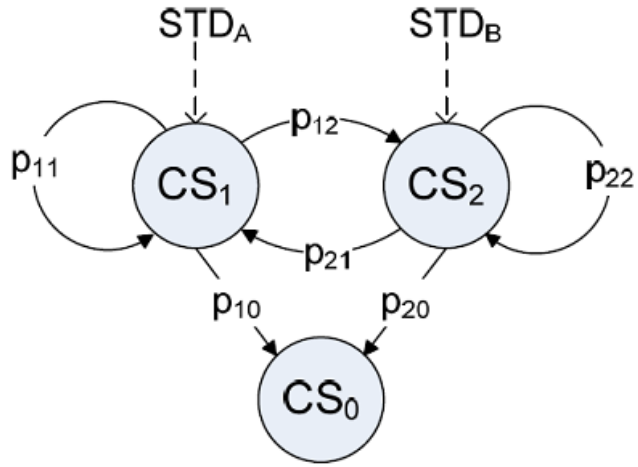
P	STD_A	STD_B
p_{11}	0.746	0.746
p_{22}	0.385	0.385
p_{12}	0.039	0.385
p_{21}	0	0.355
P_{10}	0.215	0.254
p_{20}	0.615	0.260



Perfomability Prediction: Step2-3: Values for STD_A



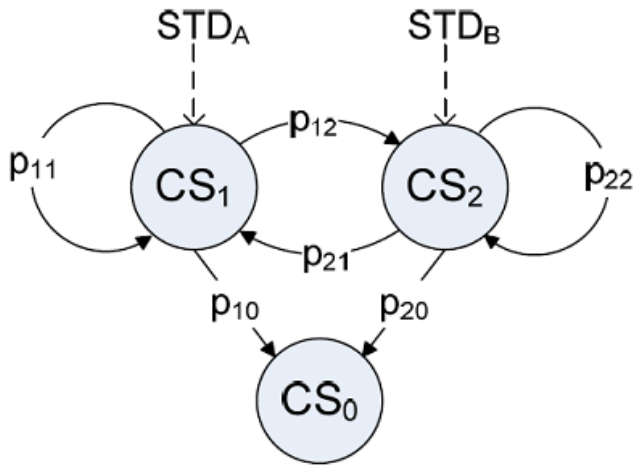
- p_{11} = 0.746 (reduction rules)
- p_{12} = 1 - R_{TMA}(1year) = 0.039
- p_{10} = 1 - p_{11} - p_{12} = 0.215
- p_{21} = 0
- p_{22} = 0.385 (reduction rules)
- p_{20} = 1 - p_{22} - p_{21} = 0.615



Perfomability Prediction: Step4: Absorbing Probabilities

- The fourth step of the algorithm calculates the absorbing probabilities $P(CS_i)$ of being in a given working configuration starting from the initial configuration CS_1 (in the STD_A case) or from the initial configuration CS_2 (in the STD_B case)

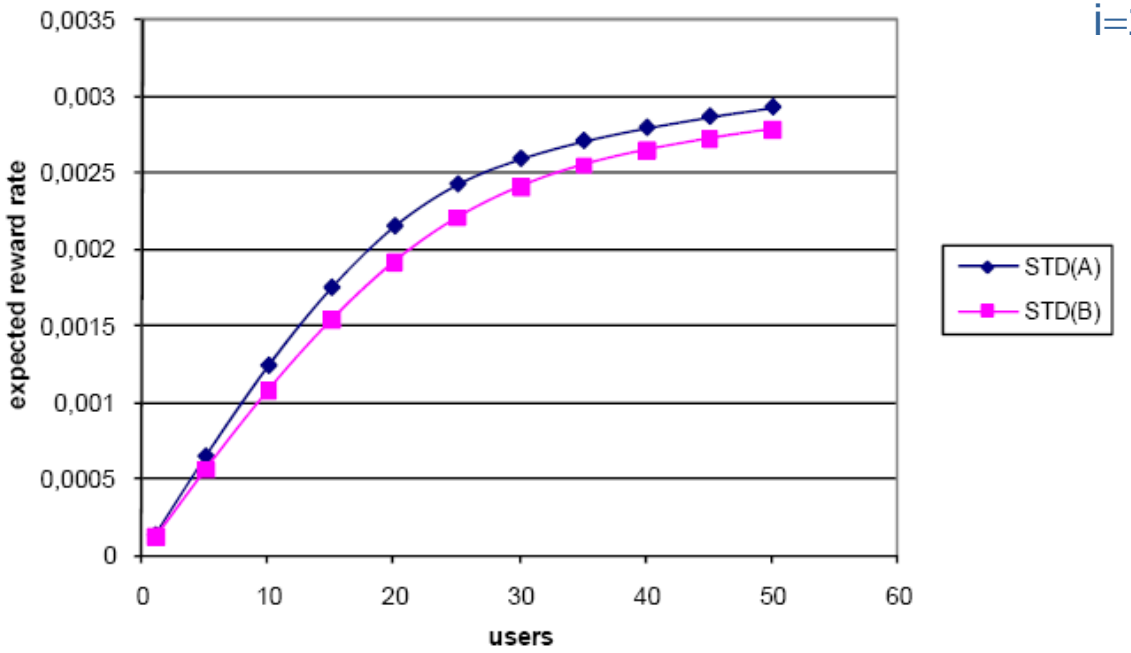
Alternative	Description
STD_A	$P(CS_1) = p_{11}$
	$P(CS_2) = p_{12} * p_{22}$
	$P(CS_0) = p_{12} * p_{20} + p_{10}$
STD_B	$P(CS_1) = p_{21} * p_{11}$
	$P(CS_2) = p_{22}$
	$P(CS_0) = p_{21} * p_{10} + p_{20}$



Perfomability Prediction: Step5-6: Expected Reward Rate

- At the fifth step the performance prediction is carried out to obtain the performance (in terms of the throughput) T(CSi) associated to each configuration
- At the sixth step, the expected reward rate is obtained by use of:

$$RW(CS) = \sum_{i=1}^{n} P(CS_i)\, T(CS_i)$$



The **performability** prediction shows that an initial configuration with $\mathbf{TM_A}$ is to be preferred, **in contrast** with what obtained from the performance prediction.

Conclusions

- Describing and predicting the QoS of composite services is a challenging and strategic issue in SOA contexts
- We have introduced a model-driven approach for predicting the performability of composite services specified by use of BPEL
- Our approach exploits an already available method for performance prediction founded on Q-WSDL
- We have introduced a model-driven method for the reliability prediction of composite services
- Such method has been then combined to the performance-related one in order to obtain a combined prediction quantified in terms of performability
- Work is in progress to implement the proposed method by use of existing performability evaluation tools

Thank You!